AMENDMENTS TO THE CLAIMS:

This listing of claims will replace all prior versions, and listings, of claims in the application:

LISTING OF CLAIMS:

1. (currently amended) A method for ~~configuring a processing system that comprises program code~~ linking program code in a processor instruction memory comprising rows and columns, the program code comprising a plurality of instructions for processing data packets in a communications network, the method comprising:

dividing the program code into a plurality of sequences, each sequence comprising a number of instructions steps and being configured to perform ~~performing~~ a certain task on a data packet passing through the communications network;~~,~~

~~providing a processor instruction memory comprising rows and columns;~~

~~allocating each sequence to a column of the processor instruction memory;~~

defining, based on the program code, a plurality of relocation objects, each relocation object of the plurality of relocation objects corresponding to a dependency relationship between two or more of the sequences;~~, and~~

allocating each sequence to at least one row and at least one column of the processor instruction memory such that the instruction steps of the sequence are consecutively allocated in the processor instruction memory; and

linking a first sequence to a second sequence by using a defined relocation object corresponding to a dependency relationship between the first sequence and the second sequence, to define a branch from the first sequence to the second sequence~~by means of the relocation object providing information that there is an alternative sequence to jump to at the instruction at which the relocation object is located~~.

2. (previously presented) The method according to claim 1, comprising the steps of forming at least one directed graph, based on at least some of the sequences and at least some of the relocation objects, and determining a longest execution path through the directed graph.

3. (previously presented) The method according to claim 2, comprising the step of entering at least one state preserving operation in the instruction memory, so as to make at least two execution paths equally long.

4. (previously presented) The method according to claim 3, comprising the step of moving at least one sequence in the instruction memory.

5. (previously presented) The method according to claim 3, wherein the length of the at least two execution paths correspond to the longest execution path.

6. (previously presented) The method according to claim 1, comprising the step of determining the existence of any circle reference by any of the relocation objects between any of the sequences.

7. (previously presented) The method according to claim 1, comprising the step of linking at least one sequence, obtained by the step of dividing the program code, to a sequence, obtained by dividing another program code.

8. (currently amended) A processing system for linking program code in a processor instruction memory comprising rows and columns~~processing data packets in a communications network~~, the processing system comprising:

an assembler comprising program code, comprising a plurality of instructions for processing data packets a ~~the~~ communications network, wherein the assembler being adapted to:

divide the program code into a plurality of sequences, each sequence comprising a number of instruction steps and being configured to perform a certain task on a data packet passing through the communications network, and

define, based on the program code, a plurality of relocation objects, each relocation object of the plurality of relocation objects corresponding to a dependency relationship between two or more of the sequences,

the processor comprising:

~~a processor instruction memory comprising rows and columns,~~

a linker being adapted to:

allocate each sequence to at least one row and at least one~~a~~ column of the processor instruction memory, such that the instruction steps of the sequence is consecutively allocated in the processor instruction memory; and

linking a first sequence to a second sequence by using a defined relocation object, corresponding to a dependency relationship between the first sequence and the second sequence to define a branch from the first sequence to the second sequence~~wherein the relocation object provides information that there is an alternative sequence to jump to at the instruction at which the relocation object is located~~.

9. (previously presented) The processing system according to claim 8, wherein the assembler is adapted to form at least one directed graph, based on at least some of the sequences and at least some of the relocation objects, and the linker is adapted to determine a longest execution path through the directed graph.

10. (previously presented) The processing system according to claim 9, wherein the linker is adapted to enter at least one state preserving operation in the instruction memory, so as to make at least two execution paths equally long.

11. (previously presented) The processing system according to claim 10, wherein the linker is adapted to move at least one sequence in the instruction memory.

12. (previously presented) The processing system according to claim 10, wherein the length of the at least two execution paths correspond to the longest execution path.

13. (previously presented) The processing system according to claim 8, wherein the linker is adapted to determine the existence of any circle reference by any of the relocation objects between any of the sequences.

14. (previously presented) The processing system according to claim 8, wherein the linker is adapted to link at least one sequence, obtained by dividing the program code, to a sequence, obtained by dividing another program code.

15. (new) The method according to claim 1, wherein the allocating each sequence to at least one row and at least one column of the processor instruction memory such that the instruction steps of the sequence are consecutively allocated in the processor instruction memory is performed by allocating each instruction step of the sequence to the same row but in different columns of the processor instruction memory.

16. (new) The processing system according to claim 8, wherein the allocating each sequence to at least one row and at least one column of the processor instruction memory is performed by allocating to the same row but in different columns.